

```
%%%%%%%%%%%%%%%
% cytosim_REF_mm.m
% This script performs the acute toxic classification method as outlined
% by OECD guidelines. This specific implementation is for:
%      "True" LD50 values from: REF
%      Estimated log10(IC50) mean and deviation units: mmol/L
%%%%%%%%%%%%%%%
close all; clear all;

% Define work space.
workDir = 'C:\Documents and Settings\jmatthews\My Documents\Dose Response Simulation';
dataDir = 'C:\Documents and Settings\jmatthews\My Documents\Dose Response
Simulation\data';
resultDir = 'C:\Documents and Settings\jmatthews\My Documents\Dose Response
Simulation\results2\REF_mm';

path(path,workDir)
path(path,dataDir)
path(path,resultDir)

% Load appropriate data set.
load Data_REFLD50_051006

% Set seed.
seed = 934502;
rand('state', seed);
randn('state', seed);

% Set fixed parameters.
maxIter = 2000; % number of iterations for each trial
sigmaArray = [0.12 0.25 0.5 1.25 2];
doseArray_mg_kg = [5 50 300 2000];
log_doseArray_mg_kg = log10(doseArray_mg_kg);
Results = [];

cd(resultDir)

for chemInd = 1:length(Data_REFLD50)

    TestData = Data_REFLD50(chemInd);

    for sigmaInd = 1:length(sigmaArray)

        sigma = sigmaArray(sigmaInd);

        for iter = 1:maxIter

            for cellInd = 1:2
```

```

if ~isnan(TestData.cellType(cellInd).est_log_ic50)
    % Choose a logged IC50 value by randomly sampling from the normal
distribution where
    % the mean is stored in the data as "est_log_ic50" and the standard
    % deviation is stored in the data as "est_log_ic50_std".
    % Use this one if we are using (mM) unit IC50 estimate.
    est_logIC50 = TestData.cellType(cellInd).est_log_ic50 + ...
        (TestData.cellType(cellInd).est_log_ic50_std * randn(1));

    % Use conversion equation based on cell type.
    if cellInd == 1 % looking at 'NHK' cell type
        est_logLD50 = 0.4519 + 0.5103 * est_logIC50; % model for REF,
(mM) case
    else % looking at 'ST3' cell type
        est_logLD50 = 0.4247 + 0.5895 * est_logIC50; % model for REF,
(mM) case
    end

    % Use this one if we are using the (mM) unit IC50 estimate.
    est_LD50 = TestData.molWeight_g_mol * (10 ^ est_logLD50);

    % Get "true" LD50 information from data.
    true_LD50 = TestData.true_ld50_mg_kg;
    true_logLD50 = log10(true_LD50);

    % Identify the starting dose by rounding down the est_LD50 value to
the
    % nearest dosing amount available.
    % NOTE: We can try other rounding schemes than this one.
    if est_LD50 < doseArray_mg_kg(1)
        dose = doseArray_mg_kg(1);
        doseIndex = 1;
    else
        index = doseArray_mg_kg < est_LD50;
        dose = max(doseArray_mg_kg(index));
        doseIndex = find(doseArray_mg_kg == dose); % get index in dose
array of start dose
    end

    % Run algorithm and find category. (for a dose group of three
animals)
    temp = toxClassAlg(doseIndex, log_doseArray_mg_kg, true_logLD50,
sigma, 3);

    % Save some more information about trial.
    temp.startDose = dose;
    temp.estLD50 = est_LD50;

```

```
temp.trueLD50 = true_LD50;

Results(chemInd).cellType(cellInd).sigma(sigmaInd).runs( iter) = temp
;
end % End if statement.

end % End cell type loop.

end % End trial iteration loop.

end % End sigma loop.

% Save data.
fName = ['cytoResults_REF_mM_' num2str(chemInd)]; % used for REF, (mM) case
save(fName, 'Results')
clear Results

end % End chemical loop.
```